Remarks

Claims 1 has been amended to incorporate recitations similar to those originally contained in dependent claim 4. More particularly, claim 1 has been amended to recite that the characters are converted "using a single hardware instruction capable of processing a plurality of characters". Claim 4 has in turn been amended to recite more particularly that the hardware instruction converts "a source string of characters encoded according to said source encoding scheme" to "a target string of characters encoded according to one of said sub-codepages", as is true of the S/390 Translate Two to One (TRTO) instruction described in the specification.

Claim 4 originally recited that "more than one character" was processed by "a single hardware instruction". Gonzalez et al. 6,204,782 ("Gonzalez") discloses a Unicode conversion scheme (Fig. 7) similar in some respects to applicant's system, but processing characters serially in a loop containing elements 704, 706, 708, 728 and 730. In rejecting original claim 4 under 35 U.S.C. § 103 as being unpatentable over Gonzalez (page 6), the Examiner contended that "[p]arallel processing yields obvious benefits in that more work is done in a give time" and that it would have been obvious "to have implemented a parallel processing algorithm to have processed more than one character in a single hardware instruction to have converted the source string faster". Applicant respectfully disagrees.

For the record, claim 1 as currently amended, as well as claim 4 as originally presented, does not claim "parallel processing" as such. Rather, it claims the use of a single hardware instruction capable of processing a plurality of characters (either serially or in parallel, depending on the implementation). As for the advantages of parallel processing, these are obvious, but only for processes that by their nature are susceptible to parallel processing. Such processes would include, for example, adding two vectors $x = (x_1, x_2, \ldots)$ and $y = (y_1, y_2, \ldots)$ to produce $x + y = (x_1 + y_1, x_2 + y_2, \ldots)$. In such a case, each individual addition operation (e.g., $x_1 + y_1$) is independent of the others, lending the operations to being parallelized. In the present case, however, each traverse of Gonzalez's loop containing elements 704, 706, 708, 728 and 730 is dependent on at least two testing operations (at steps 708 and 730). Thus it is not at all obvious, looking just at the Gonzalez reference, just how such a loop could be speeded up or even whether it is amenable to being speed up.

For the foregoing reasons, applicant respectfully submits that claim 1 as amended and the claims dependent thereon distinguish patentably over the art cited by the Examiner.

Claim 10 has been amended to recite that the computer program is "on a computer-readable medium" and comprises "program code" for performing respective steps of the method if claim 1. Accordingly, claim 10 and claim 11 dependent thereon are believed to overcome the Examiner's rejections based upon 35 U.S.C. §§ 112 and 101 (pages 2-3).

New claim 14 is directed to that feature of applicant's invention described at page 12, lines 4-8, whereby the highest-priority sub-codepage, rather than the sub-codepage found to contain the character, is made the new current sub-codepage if the found sub-codepage belongs to a certain subclass of codesets. Claim 14 is generally modeled on original claim 1, but further characterizes the conversion process. More particularly, the sub-codepages comprise a first set of one or more higher-priority sub-codepages and a second set of one or more lower-priority sub-codepages. Initially, the highest-priority sub-codepage is selected as a current sub-codepage, and the conversion process take place using the current sub-codepage.

If a character is encountered that is not contained in the current sub-codepage, then the other sub-codepages are searched in priority order for a sub-codepage containing the character. If the character is found in a sub-codepage belonging to the first set of sub-codepages (i.e., the higher-priority sub-codepages), the conversion process is continued with that sub-codepage as the current sub-codepage. On the other hand, if the character is found in a sub-codepage belonging to the second set of sub-codepages, the character is converted using that sub-codepage and the conversion process is continued with the highest-priority sub-codepage as the current sub-codepage.

Claims 15-20, dependent on claim 14, are based upon selected ones of claims 2-13.

Claims 14-20 are believed to distinguish patentably over the art cited by the Examiner and, in particular, are believed to distinguish patentably over Gonzalez. The method recited in claim 14 combines the best features of Gonzalez's first approach of always reverting to the highest-priority sub-codepage and his second approach of continuing with the sub-codepage in which a character is found (col. 6, lines 44-54; col. 15, lines 13-31. Normally, continuing with the sub-codepage in which a character is found is an optimal approach because of the principle of locality, whereby adjacent characters are likely to belong to the same sub-codepage. However, some sub-codepages, such as applicant's codesets 2 and 3, are so rare in occurrence that this principle does not apply. Blindly making such a rarely encountered codeset the new current codeset for a conversion process will result in a great deal of unnecessary overhead as, predictably, the next character is not found in that codeset. Applicant's claimed method, however, avoids this outcome.

For the foregoing reasons, applicant respectfully submits that new claim 14 and the claims dependent thereon likewise distinguish patentably over the art cited by the Examiner.

Reconsideration of the application as amended is respectfully requested. It is hoped that upon such consideration the Examiner will hold all claims allowable and pass the case to issue at an early date. Such action is earnestly solicited.

Respectfully submitted,
JOACHIM M. BAUER

By

William A. Kinnaman, Jr.
Registration No. 27,650
Phone: (845) 433-1175
Fax: (845) 432-9601

WAK/wak